Proceedings of the Sixth International Conference on Machine Learning and Cybernetics, Hong Kong, 19-22 August 2007

ENHANCED ALGORITHM PERFORMANCE FOR CLASSIFICATION BASED ON HYPER SURFACE USING BAGGING AND ADABOOST

QING HE[1], FU-ZHEN ZHUANG[1,2], XIU-RONG ZHAO[1,2], ZHONG-ZHI SHI[1]

[1]The Key Laboratory of Intelligent Information Processing, Department of Intelligence Software, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, 100080, China
[2]Graduate University of Chinese Academy of Sciences, Beijing 100039, China
E-MAIL: {heq, zhuangfz, zhaoxr, shizz}@ics.ict.ac.cn

Abstract:

To improve the generality ability of Hyper Surface Classification (HSC), Bagging and AdaBoost ensemble learning methods are proposed in this paper. HSC is a covering learning algorithm, in which a model of hyper surface is obtained by adaptively dividing the sample space and then the hyper surface is directly used to classify large database based on Jordan Curve Theorem in Topology. Experiments results confirm that Bagging and AdaBoost can improve the generality ability of Hyper Surface Classification (HSC) in general. However, its behavior is subject to the characteristics of Minimal Consistent Subset for a disjoint Cover set (MCSC). Usually the accuracy of Bagging and AdaBoost can not exceed the accuracy predicted by MCSC. So MCSC is the backstage manipulator of generalization ability.

Keywords:

Minimal consistent subset; Hyper surface cassification; Bagging; AdaBoost

1. Introduction

Hyper Surface Classification (HSC) is a novel classification method based on hyper surface is put forward by He & Shi & Ren [1]-[4]. In this method, a model of hyper surface is obtained by adaptively dividing the sample space in the training process and then the hyper surface is directly used to classify large database according to whether the wind number is odd or even based on Jordan Curve Theorem in Topology. Some covering learning algorithms were proposed in the decade time. Bionic (Topological) Pattern Recognition theory is firstly proposed by Wang Shoujue as a new model for pattern recognition [5]. BPR is based on "matter cognition" instead of "matter classification", and so is thought closer to the function of human cognition than traditional statistical pattern recognition using "optimal separating" as its main principle. HSC, also a kind of cover algorithm, has the common goal of feature cognition as BPR, but the hyper surface is obtained by adaptively dividing the sample space in the training process and the classification method is based on the Jordan Curve Theorem. Experiments show that HSC can efficiently and accurately classify large-size data in two-dimensional space and three-dimensional space [1] [2]. Though HSC can classify higher dimensional data according to Jordan Curve Theorem on the theoretical plane, it is not as easy as in three-dimensional space. However, what we really need is an algorithm that can deal with data not only of massive size but also of high dimensionality. Thus in [3] a simple and effective kind of dimension reduction method without losing any essential information is proposed. Formally, the method is a dimension reduction method but in nature the method is a dimension transposition method. Moreover, we proposed another different method based on ensemble in paper [4]. The most important difference between HSC ensemble and the traditional ensemble is that the sub-datasets are obtained by dividing the features rather than by dividing the sample set. Experiments show that this method has a preferable performance on high dimensional data sets, especially on those in which samples are different in each slice [4]. For large density three-dimensional data up to 10^7 the speed of HSC is still very fast [2]. For small sparse data set, HSC is not so good. This is the motivation of this paper.

Previous studies have shown that aggregating ensemble classifiers is often more accurate than a single classifier in the ensemble [6]–[7]. Bagging and Boosting are two popular methods that have shown success in a great variety of data domains [8], [9]. Bagging forms bootstrap replicates by random sampling, with replacement, from the original data set. Boosting attaches weight to each instance at each repetition so that the ensemble classifier focuses on harder instances. After creating classifier replicates, both approaches aggregate the votes from multiple classifiers. In bagging, the vote of each ensemble classifier carries the same weight, while the vote in boosting is weighted by the

1-4244-0973-X/07/$25.00 ©2007 IEEE

relative accuracy performance of each classifier. To improve the generality ability of Hyper Surface Classification (HSC)，Bagging and AdaBoost ensemble learning methods are proposed.

The rest of this paper is organized as follows: In section 2, we give an outline of Hyper Surface Classification (HSC). Then in Section 3 the concept and construction of Minimal Consistent Subset for a disjoint Cover set (MCSC) of HSC are described. In section 4 and section 5 How to enhance algorithm performance for Classification Based on Hyper Surface by using Bagging and Boosting is described. In Section 6, experimental results are presented, followed by our conclusions in Section 7.

2. Overview of the classification method based on hyper surface

HSC is a universal classification method based on Jordan Curve Theorem in topology.

Jordan Curve Theorem. Let X be a closed set in n-dimensional space R^n. If X is homeomorphic to a sphere in $n-1$ dimensional space, then its complement $R^n \setminus X$ has two connected components, one called inside, the other called outside.

Classification Theorem. For any given point $x \in R^n \setminus X$, x is in the inside of $X \Leftrightarrow$ the wind number i.e. intersecting number between any radial from x and X is odd, and x is in the outside of $X \Leftrightarrow$ the intersecting number between any radial from x and X is even.

How to construct the separating hyper surface for any data set, which distribution is unknown, is an important problem. Based on Jordan Curve Theorem, we have put forward the basic classification method based on separating hyper surface in [1] [2]. Numerical examples of problems with three-dimensional data will be trained and classified. The following is the general Hyper Surface Classification method for k classes d-dimensional data.

Step1. Input the training samples, which are composed of k classes d-dimensional data. Let the training samples be distributed within the rectangle region.

Step2. Transform the region into a unit region.

Step3. Divide the region into $\overbrace{10\times10\times\cdots10}^{d}$ small regions.

Step4. Label the small regions by $1,2,\cdots,k$ according to whether the samples' class in the region is only Class 1, Class 2, or Class k, respectively.

Step5. Remerge the frontiers of the same class regions, which labeled is the same, then save it as a link table.

Step6. For the small regions, where there is more than one class of sample then repeat Step3 to step5.

Step7. Input a testing sample and make a radial from the sample.

Step8. Input all the link tables obtained in the above training process.

Step9. Count the number of intersections of the sample with the above link table.

Step10.If the number of intersections of the sample with some link table is odd then label the sample class the same as the link table, otherwise go to next link table.

Step11. If the number of intersections of the sample with all above link table is even then the sample's class cannot be defined.

Step12. Calculate the classification accuracy.

The classification algorithm based on hyper surface is a polynomial algorithm when the same class samples are distributed in finite connected components. Experiments show that HSC can efficiently and accurately classify large density data in two-dimensional or three-dimensional space for multi-classification. For small sparse data set, HSC is not so good. In the following we can see MCSC the backstage manipulator of generalization ability.

3. Minimal consistent subset for a disjoint cover set

For selecting a representative subset of the original training data, Minimal Consistent Subset (MCS) was presented by Hart in 1968[14]. The detail about the study of MSC can be seen in paper [15]. Minimal Consistent Subset (MCS) is a consistent subset with a minimum number of elements. Every set has a consistent subset, since every set is trivially a consistent subset of itself. Obviously, every finite set has a minimal consistent subset, although the minimum size is not, in general, achieved uniquely. We define Minimal Consistent Subset for a disjoint Cover set (MCSC) of HSC as following.

Suppose C is all subsets collection of a finite sample set S. A disjoint cover set for S, i.e., a subset $C' \subseteq C$ such that every element in S belongs to one and only one

member of C'. Minimal Consistent Subset for a disjoint cover set (MCSC) C' is a sample subset combined by choosing one and only one sample from each subset in the disjoint cover set C'. A disjoint cover set C' can be constructed by using HSC. The disjoint cover set C' is the union set of samples subsets in all of the units that are included in the hyper surface H. Let $\overline{H}$ be the interior of H. Let u be a unit in H. Minimal Consistent Subset based on hyper surface denoted by $S_{\min}|_H$ is a sample subset combined by selecting one and only one representative sample from each unit included in the hyper surface, i.e.

$$S_{\min}|_H = \bigcup_{u \subseteq \overline{H}} \{\text{choosing one and only one } s \in u\}$$

We call sample a and b are equivalent when they are with the same category and fall into the same unit.

For any given sample set, how to construct the Minimal Consistent Subset based on hyper surface is very important. The following is its detail steps:

Step1. Let the given samples distribute in the inside of a rectangle region.

Step2. Divide the region into some smaller regions called units, which only contain at most one class of sample. If there exist some smaller regions where there are more than one class of sample then repeat Step2 in these smaller regions until each small region only contains at most one class of sample.

Step3. Label each region according the inside samples' class. Then the frontier vectors and the class vector form a string.

Step4. Combine the adjacent region of the same class and obtain a separating hyper surface then save it as a string.

Step5. Import the separating hyper surface into memory, during which the layers these pieces of hyper surface lie in are remembered.

Step6. For each sample in the training set, locate its position in the model, which means to figure out which unit it locates in.

Step7. Combine samples that locate in the same unit into one equivalent class, then we get a number of equivalent classes in different layers.

Step8. Pick up one sample from each equivalent class to form the MCSC about the separating hyper surface.

From the algorithm above, we can see that the number of samples in each Minimal Consistent Subset equals to the number of equivalent classes. And the number of MCSC equals to the size of Cartesian product of these equivalent classes. The method indeed ensures consistency and minimal for given cover set by hyper surface. Moreover, it is not sensitive to the randomly picked initial selection and to the order of consideration of the input samples.

To illustrate the concept of MCSC based on HSC, we list the following two figures. Figure1 shows the separating hyper surface of the data set of breast-cancer-Wisconsin, which contains 699 samples, firstly changed into 3 dimensional data by using the dimension transposition based on ranking method in section 3. And the separating hyper surface of its Minimal Consistent Subset obtained by using the algorithm in Section 3, which contains 229 samples, is shown in Figure2.

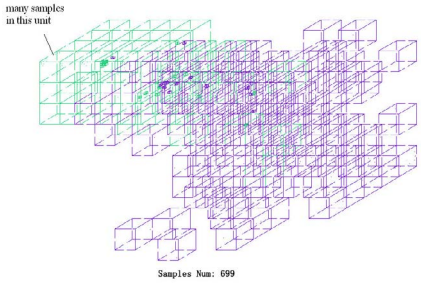

Figure 1. The hyper surface of the data set of breast-cancer-wisconsin

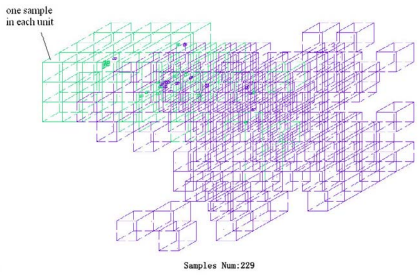

Figure 2. The hyper surface of MCSC for breast-cancer-wisconsin

4. Bagging

A learning algorithm is unstable for small data set if small changes in the training data set will generate very diverse classifiers. Breiman [8] proposed the use of bagging to improve performance by taking advantage of this effect. A learning algorithm combination is informally called unstable if "small" changes in the training data lead to significantly different classifiers and relatively "large" changes in accuracy. In general, bagging improves recognition for unstable classifiers because it effectively averages over such discontinuities. However, there are no convincing theoretical derivations or simulation studies showing that bagging will help all unstable classifiers.

The implement procedures of Bagging Algorithm are

described in details as follows:

Step1. First we get the training data set D which has N samples, then set the cycle times K and selected number $n(n < N)$ when we do bagging.

Step2. Sampling randomly from data set D and receiving a new data set D_k. For any given algorithm, a general classifier C_k will be generated from D_k.

Step3. Repeat Step2 K times and we will get K classifiers.

Step4. When comes a test sample, the final classification decision is base on the vote of above gained classifiers.

5. Boosting and adaboost

Boosting generates new classifier ensembles by readjusting the weight attached to each instance in a way that new ensemble classifiers will focus on difficult cases. The training set for each ensemble depends on the performance of previous classifier(s). AdaBoost Among all the theoretically provable boosting techniques, the most successful one in practical applications has been AdaBoost due to Freund and Schapire [10]. The explanation of its success comes from two reasons, first its simplicity and seconds a property of AdaBoost that previous boosting algorithms [13] lacked of, namely, adaptively". The algorithm adapts its strategy to the situation being used, which free its user from the difficulty of determining algorithmic parameters.

In AdaBoost each training pattern receives a weight that determines its probability of being selected for a training set for an individual component classifier. If a training pattern is accurately classified, then its chance of being used again in a subsequent component classifier is reduced; conversely, if the pattern is not accurately classified, then its chance of being used again is raised. In this way, AdaBoost "focuses in" on the informative or "difficult" patterns. Specifically, we initialize the weights across the training set to be uniform. On each iteration *k*, we draw a training set at random according to these weights, and then we train component classifier C_k on the pattern selected. Next we increase weights of training patterns misclassified by C_k and decrease weights of the patterns correctly classified by C_k. Patterns chosen according to this new distribution are used to train the next classifier C_{k+1}, and the process is iterated.

Here you can see the general description of AdaBoost Algorithm using the baseline algorithm HSC.

Step1. Input N patterns $\{(x_1, y_1),...,(x_N, y_N)\}$, and assign each pattern the same weight which determines its probability of being selected for a training set, calling the initial pattern distribution $D^{(1)}$.

Step2. Train basic classifier HSC according to the pattern distribution $D^{(t)}$ and attain classifier C_t.

Step3. Compute the pattern misclassified rate, using the following expression:

$$\varepsilon_t = \sum_{n=1}^{N} D_n^{(t)} I(y_n \neq C_t(x_n))$$

Step4. Compute weight of the corresponding classifier with the following expression:

$$\alpha_t = \frac{1}{2} \log \frac{1 - \varepsilon_t}{\varepsilon_t}$$

Step5. Renew the weight of each pattern according to the following expression:

$$D_n^{(t+1)} = D_n^{(t)} \exp(-\alpha_t y_n C_t(x_n)) / Z_t$$

Where Z_t is normalizing constant:

$$Z_t = \sum_{n=1}^{N} D_n^{(t)} \exp(-\alpha_t y_n C_t(x_n))$$

Step6. Do the circle step2~5 T times, we will attain T classifiers. The final ensemble classifier:

$$f_{Ens}(x) = \sum_{t=1}^{T} \alpha_t C_t(x)$$

6. Experiments

As we describe above, our purpose in this paper is to enhance algorithm performance for classification based on hyper surface by using Bagging and AdaBoost. Exciting result will be achieved in the following experiment. In this experiment we use Bagging algorithm with the basic algorithm HSC, at last the results are compared with the method of using HSC merely. We select n different samples randomly from the sample set of breast-cancer-wisconsin and the rest used for test and the results are shown in Table.1. From the results we can clearly find that only minority samples for training, very good performance is attained when using Bagging algorithm. For example, only 40 samples are used in every cycle can get the accuracy up to 90%.This is very useful for that if we have not enough samples for training, because we can maintain the good performance of classification in some certain. From the result, we also find that the test accuracy falls down in generally with the number of training samples decreasing. Therefore the number of training samples influences the performance of HSC and Bagging.

To illustrate the importance of the number 229 which is the size of Minimal Consistent Subset for a disjoint Cover set (MCSC) in the data set of breast-cancer-Wisconsin, the following experiments have been done. The

experiment result shown in Table.2. In the experiment $n(n = 29, 79, 129, 229, 329, 429)$ different numbers of samples from 699 are selected and the selected samples were trained by Bagging algorithm. The rest samples $699 - n$ are tested, and the accuracy is shown in the corresponding table.

Table 1. n samples selected from samples set of breast-cancer-wisconsin (699 samples)

Samples Selected	200	180	160	120	100	80
HSC Accuracy	67.13%	66.28%	64.38%	63.73%	61.43%	58.32%
Bagging Accuracy	98.20%	98.27%	98.14%	97.93%	97.66%	95.96%
Samples Selected	60	50	40	30	20	10
HSC Accuracy	55.71%	54.08%	51.29%	54.71%	45.21%	36.28%
Bagging Accuracy	96.71%	95.07%	91.50%	91.18%	85.27%	77.36%

Table 2. n samples are selected from 699, and the rest are tested

	29	79	129	229	329	429
1	88.06%	96.94%	97.72%	98.09%	98.65%	98.15%
2	88.66%	97.42%	97.72%	98.51%	97.57%	98.15%
3	89.40%	97.26%	97.54%	98.94%	98.92%	97.07%
4	89.85%	96.45%	97.37%	98.30%	97.84%	99.26%
5	89.70%	97.26%	97.54%	98.09%	97.84%	99.26%
6	91.04%	97.26%	97.19%	97.23%	98.65%	98.89%
7	89.40%	96.77%	97.19%	98.51%	98.38%	98.52%
8	90.15%	97.90%	97.37%	97.87%	97.84%	99.63%
9	88.96%	97.58%	97.89%	97.87%	98.65%	97.78%
10	91.19%	97.58%	97.37%	98.72%	98.92%	98.52%
Average Accuracy	89.64%	97.24%	97.49%	98.21%	98.32%	98.52%

From Table.2 we find that with the selected number increasing, the corresponding accuracy rises. But when n is up to 229(the size of the MCSC), the nearly best average accuracy is got. Though we increase the number of training samples drastically to 329 and 429, we can not improve the performance of classification obviously. It seems that when we sampling 229 samples, we nearly get all the information of total sample set. So the size of MCSC is a key number when we select samples for training. Next experiment will enforce the importance of MCSC, and it seems that the size of MCSC constrains the performance of classification of Bagging and Adaboost algorithm. In this experiment we pick up a sample from the MCSC selectively as training sample set (the rest 699-299+1 samples for tested) and calculate the accuracy using the algorithm HSC, Bagging and AdaBoost.

Table 3. Single deletion from the minimum sample set of breast-cancer-wisconsin

Samples in the same unit with the one deleted	ID of deleted sample	HSC Accuracy by Experiment	Bagging Accuracy by Experiment	AdaBoost Accuracy by Experiment
1	4	99.79%	99.79%	99.79%
2	26	99.58%	99.36%	99.58%
3	10	99.36%	99.15%	99.36%
4	27	99.15%	98.94%	99.15%
5	35	98.94%	98.73%	98.73%
6	43	98.73%	98.73%	98.73%
7	20	98.51%	98.51%	98.30%
8	30	98.30%	98.30%	98.09%
10	6	97.88%	97.66%	97.66%
11	178	97.66%	97.45%	97.45%
17	37	96.39%	96.39%	96.18%
34	17	92.78%	100%	99.79%
39	9	91.72%	91.51%	91.72%
48	1	89.81%	89.81%	89.81%
71	3	84.93%	84.93%	84.93%
117	7	75.16%	75.16%	74.95%

The result in Table.3 show that the accuracy of Bagging and AdaBoost can hardly higher than HSC. This confirms the conclusion that the size of MCSC constrains the performance of classification of Bagging and Adaboost. Its behavior is subject to the characteristics of Minimal Consistent Subset for a disjoint Cover set (MCSC).

Usually the accuracy of Bagging and AdaBoost can not exceed the accuracy predicted by MCSC. But Bagging and AdaBoost can also guarantee the accuracy without being influenced by outliers. For example, there is an outlier, when sample 17 is deleted from the training set, whose accuracy of Bagging and AdaBoost is drastically higher than HSC by approximately $\xi = 7.22\%$. After analyzing the cause, we find that when using HSC, the representative sample is deleted, so the samples in the same unit with the one deleted can not be classified accurately. We compute the misclassified rate only in this unit, 34/471=7.22% which conforms to ξ . Why samples in the same unit with one deleted can be also classified accurately when using Bagging algorithm? We will explain it in Fig.3. Fig.3 is a simple illustration, actually the case is more complicated. Given unit L is the unit which contains the representative sample of ID 17. When the sample of ID 17 is deleted, the type of the Unit L becomes unknown, so its type dependents on its neighbouring units. Because of the randomly sampling of Bagging, if the selected data set does not contain Unit J or Unit K, the Unit L may be classified accurately by the weak classifier. When combining the adjacent region of the same class, the Unit L will be labeled with "+" if the representative sample of Unit J and K are not selected. There is one other condition that the Unit J and K contain only very few samples. So in this special case of

data distribution, the Bagging and Adaboost can do better performance.

Unit I (+)	Unit J(-)
Unit K(-)	Unit L (+)

Figure.3.The sketch map of four units

Table 4. 229 samples randomly selected from 699 (no MCSC) for training, and the rest samples 699-229 are tested

	HSC by Experiment	Bagging Accuracy by Experiment	AdaBoost Accuracy by Experiment
1	64.89%	64.89%	64.89%
2	64.89%	65.11%	64.68%
3	64.89%	65.32%	65.74%
4	64.89%	65.11%	64.47%
5	64.89%	65.32%	65.53%

Another experimental result shown in Table.4 is that we select 229 samples randomly from 699 (no MCSC) for training, and the rest $699-229$ samples for testing. From Table.4, we can see that if the training set does not contain MCSC, the accuracy is not good, much lower than the MCSC. These show that MCSC is the backstage manipulator of generalization ability.

7. Conclusions

To improve the generality ability of HSC, Boosting and AdaBoost algorithm are used to improve and complement the performance of HSC. Experiments results confirmed that Bagging and AdaBoost can improve the generality ability of Hyper Surface Classification (HSC) in general. It is exciting that the result is good, and specially, the size of training set is sharply reduced with the accuracy decreasing unremarkably. However, its behavior is subject to the characteristics of Minimal Consistent Subset for a disjoint Cover set (MCSC). Usually the accuracy of Bagging and AdaBoost can not exceed the accuracy predicted by MCSC. So MCSC is the backstage manipulator of generalization ability.

Aknowledgements

This work is supported by the National Science Foundation of China (No. 60435010, 90604017, 60675010), 863 National High-Tech Program (No.2006AA01Z128), National Basic Research Priorities Programme (No. 2003CB317004) and the Nature Science Foundation of Beijing (No. 4052025).

References

[1] Q. He, Z. Z. Shi, L. A. Ren, "The classification method based on hyper surface, "IJCNN '02. Proceedings of the 2002 International Joint Conference on Neural Networks, pp.1499-1503, 2002.

[2] Q. He, Z. Z. Shi, L. A. Ren, E. S. Lee, "A Novel Classification Method Based on Hyper Surface, " International Journal of Mathematical and Computer Modeling, pp. 395-407, 2003.

[3] Q. He, X. R. Zhao, Z. Z. Shi, "Classification based on dimension transposition for high dimension data," SOFT COMPUTING 11(4) pp.329-334, 2007.

[4] X. R. Zhao, Q. He, Z. Z. Shi, "HyperSurface Classifiers Ensemble for High Dimensional Data Sets, " In Wang et al. (Eds.): 3rd International Symposium on Neural Networks (ISNN 2006), LNCS 3971, pp. 1299 – 1304, 2006.

[5] S. J. Wang, "Bionic (Topological) Pattern Recognition-A New Model of Pattern Recognition Theory and Its Applications，" ACTA ELECTRONICA SINICA 30(10), pp. 1417-1420L, 2002.

[6] K. Hansen, P. Salamon, "Neural Network Ensembles," IEEE Transaction on Pattern Analysis and Machine Intelligence, 12(10), pp. 993-1001, 1990.

[7] Z.H. Zhou, J. Wu, Y.,Jiang, S.F. Chen, "Genetic Algorithm based Selective Neural Network Ensemble, " International Joint Conference on Artificial Intelligence, pp. 797-802, 2001.

[8] L. Breiman, "Bagging predictors," Mach. Learn., vol. 24, pp. 123–140,1996.

[9] Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm," in Proc. 13th Int. Conf.: Mach. Learn., Bari, Italy, July 3–36 1996, pp. 148–156.

[10] Y. Freund and R. E. Schapire, "A Decision-Theoretic Generalization of On-Line Learningand an Application to Boosting" Journal of computer and system sciences 55, pp.119-139, 1997.

[11] E. Bauer and R. Kohavi, "An empirical comparison of voting classification algorithms: Bagging, boosting, and variants," Machine Learning, vol. 5, pp. 1–38, 1998.

[12] R. Maclin and D. Opitz, "An empirical evaluation of bagging and boosting," in Proc. 14th Natl. Conf. Artificial Intelligence. Providence, RI, July 27–31 1997.

[13] Y. Freund, "Boosting a weak learning algorithm by majority", Inform. and Comput. 121, No. 2 (September 1995), 256_285; an extended abstract appeared in ``Proceedings of the Third Annual Workshop on Computational Learning Theory, 1990.

[14] P. E. Hart, "The condensed nearest neighbor rule", IEEE Trans. Information Theory, IT214, pp.3515-516, 1968.

[15] H. B. Zhang. & Sun, G. Y. "Optimal reference subset selection for nearest neighbor classification by tabu search", Pattern Recognition, 35 pp.1481-1490, 2002.